

L'INTELLIGENZA ARTIFICIALE NELL'ARBITRATO E NELLE PROFESSIONI: OPPORTUNITÀ, RISCHI E COMPLIANCE

Strumenti operativi e normativi per l'integrazione
dell'IA nella giustizia arbitrale

Camera di Commercio
Cagliari - Oristano



CAMERA ARBITRALE



12 dicembre 2025

**CAMERA DI COMMERCIO DI
CAGLIARI-ORISTANO**

**Largo Carlo Felice n. 72
CAGLIARI**

IA generativa e agentica nell'arbitrato.

Applicazioni pratiche e
governance.

Relazione dell'Avv. Enrica Priolo



Ordine dei Dottori Commercialisti
e degli Esperti Contabili di Cagliari



Ordine dei
Periti Industriali
di Cagliari

Socio Aderente a SF_OIC



Ordine
dei Dottori Agronomi
e dei Dottori Forestali
della Provincia di Cagliari



Ministero della Giustizia

PROGRAMMA DELL'EVENTO

Saluti istituzionali

Ing. Maurizio de Pascale - Presidente della Camera di Commercio di Cagliari-Oristano

Dott. Cristiano Erriu - Segretario Generale della Camera di Commercio di Cagliari-Oristano

Avv. Matteo Pinna – Presidente dell’Ordine degli Avvocati di Cagliari

Avv. Enrico Maria Meloni - Presidente dell’Ordine degli Avvocati di Oristano

Dott. Alberto Vacca – Presidente dell’Ordine dei Commercialisti di Cagliari

Ing. Federico Miscali – Presidente dell’Ordine degli Ingegneri della Provincia di Cagliari

Dottore Agronomo Ettore Crobu – Presidente dell’Ordine dei Dottori Agronomi e dei Dottori Forestali della Provincia di Cagliari

Presentazione del Convegno

Dott.ssa Grazia Corradini – Presidente del Consiglio e della Camera Arbitrale presso la Camera di Commercio di Cagliari-Oristano

Apertura lavori

Prof. Carlo Dore - Professore di diritto privato presso la Facoltà di scienze economiche, giuridiche e politiche dell’Università di Cagliari e Vice - Presidente del Consiglio della Camera Arbitrale presso la Camera di Commercio di Cagliari–Oristano

Avv. Enrico Maria Meloni - avvocato Cassazionista, Presidente dell’Ordine degli Avvocati di Oristano e Vice Presidente Vicario del Consiglio della Camera Arbitrale presso la Camera di Commercio di Cagliari-Oristano

Prof. Ivan Blečić - Direttore del Dipartimento in Ingegneria Civile, Ambientale e Architettura dell’Università degli Studi di Cagliari - Professore ordinario di Estimo e Valutazione presso il Dipartimento di Ingegneria Civile, Ambientale e Architettura dell’Università degli Studi di Cagliari - Intelligenze Artificiali nella valutazione e pratica estimativa.

Keynote: IA generativa e agentica nell’arbitrato. Applicazioni pratiche e governance.

Avv. Enrica Priolo – Dati e nuove tecnologie, DPO, ODV

L’IA come strumento di supporto alla decisione arbitrale. Compliance normativa, rischi e responsabilità.

Prof. Gianmarco Gometz - Professore Ordinario di Informatica Giuridica e Filosofia del diritto presso il Dipartimento di Giurisprudenza dell’Università di Cagliari

Cybersicurezza e IA nel contesto dell’arbitrato: dal threat landscape all’implementazione di security by design e resilienza.

Avv. Francesco Paolo Micozzi, avvocato cassazionista

Progettare e implementare sistemi IA affidabili. Insidie e trappole nella progettazione dei sistemi di IA con particolare riferimento all’Arbitrato.

Prof. Ing. Gian Luca Marcialis – Docente di Sistemi di Elaborazione delle Informazioni presso il Dipartimento di Ingegneria Elettrica ed Elettronica dell’Università degli Studi di Cagliari

Indice generale

IA generativa e agentica nell'arbitrato. Applicazioni pratiche e governance.....	5
La rivoluzione silenziosa.....	6
Parte I.....	7
fondamenta tecniche.....	7
1. Anatomia di un Large Language Model.....	7
1.1 <i>L'Architettura Transformer. Una rivoluzione concettuale</i>	7
<i>Il Meccanismo di attenzione, matematica e intuizione</i>	7
1.2 <i>Dalla tokenizzazione agli embedding spaces</i>	7
1.3 <i>Strati di trasformazione e rappresentazioni gerarchiche</i>	8
2. Il processo generativo: predizione stocastica e <i>hallucination</i>	9
2.1 <i>Autoregressive generation. Come gli LLM producono testo</i>	9
2.2 <i>Temperature e sampling strategies</i>	9
2.3 <i>L'hallucination problem in una prospettiva tecnica</i>	9
3. Retrieval-Augmented Generation.....	10
3.1 <i>Il Paradigma RAG: separare memoria da ragionamento</i>	10
3.2 <i>Embeddings semantici e ricerca vettoriale</i>	10
3.3 <i>RAG / Fine-tuning / In-context learning</i>	10
Parte II.....	12
Applicazioni forensi avanzate.....	12
4. Chain-of-Thought Reasoning e verificabilità.....	12
4.1 <i>Il problema del ragionamento opaco</i>	12
4.2 <i>Anatomia di un Prompt Chain-of-Thought</i>	12
4.3 <i>Zero-shot / Few-shot CoT</i>	12
5. Prompt engineering come competenza forense.....	13
5.1 <i>Il prompt come atto procedurale</i>	13
5.2 <i>Anatomia di un prompt efficace per arbitrato</i>	13

5.3 <i>Tecniche Avanzate: constitutional ai e self-critique</i>	13
6. Multi-Agent systems, il futuro della deliberazione assistita.....	14
6.1 <i>Dal modello singolo al sistema Multi-Agente</i>	14
6.2 <i>Debate e adversarial collaboration</i>	14
6.3 <i>Tool: Implementazioni Esistenti</i>	14
Parte III.....	15
Governance epistemica e controllo.....	15
7. L'Arbitro come Human-in-the-Loop.....	15
7.1 <i>Il paradigma HITL che va oltre la supervisione passiva</i>	15
7.2 <i>Framework di Validazione Epistemica</i>	15
8. Adversarial testing. Red-Teaming dei sistemi IA.....	16
8.1 <i>Il concetto di Red-Teaming</i>	16
8.2 <i>Casi di Studio: failure modes documentati</i>	16
9. Tracciabilità, auditabilità e governance operativa.....	17
9.1 <i>Il problema del non-determinismo</i>	17
9.2 <i>Protocollo di documentazione completa</i>	17
9.3 <i>Disclosure e trasparenza verso le parti</i>	17
Verso un'intelligenza ibrida.....	18
Bibliografia Selezionata.....	19

Keynote

IA generativa e agentica nell'arbitrato. Applicazioni pratiche e governance.

Relazione dell'Avv. Enrica Priolo



Avvocato che opera nel campo delle nuove tecnologie, data protection, criminalità informatica, privacy, sicurezza informatica e infosec; compliance integrata; specializzata in diritti umani. Responsabile della protezione dei dati in aziende private e pubbliche. Formatrice esperta presso numerosi istituti privati e pubblici. Cultrice di materia presso la cattedra di Informatica giuridica, DIEE, Università di Cagliari.

L'ARBITRATO NELL'ERA DEI LARGE LANGUAGE MODELS

Dalla teoria dei trasformatori alla pratica forense

Un'analisi tecnica dell'intelligenza artificiale generativa
per la pratica arbitrale contemporanea

Avv. Enrica Priolo

"The question of whether a computer can think is no more interesting than the question of whether a submarine can swim."

— Edsger W. Dijkstra

LA RIVOLUZIONE SILENZIOSA

Nel giugno 2017, un gruppo di ricercatori di Google pubblicò un paper che avrebbe cambiato per sempre il panorama dell'intelligenza artificiale: "*Attention Is All You Need*" (Vaswani et al., 2017). L'architettura Transformer ivi proposta non era semplicemente un'innovazione incrementale, ma un cambio di paradigma che avrebbe reso possibile, sei anni dopo, sistemi come GPT-4, Claude 3.5 Sonnet e Gemini Advanced.

Oggi, questi Large Language Models (LLM) stanno penetrando silenziosamente ogni ambito della pratica legale. Ma mentre proliferano gli articoli divulgativi che celebrano o demonizzano l'IA, resta una lacuna fondamentale: la comprensione tecnica di *come* questi sistemi funzionano realmente, e *perché* le loro capacità e limitazioni hanno implicazioni profonde per l'arbitrato.

Questa relazione si propone di colmare tale lacuna. Non ci limiteremo a descrivere *cosa* l'IA può fare per l'arbitrato, ma esploreremo le fondamenta architettoniche, i meccanismi cognitivi sottostanti, e le implicazioni epistemologiche dell'uso di sistemi stocastici in contesti che richiedono certezza giuridica.

Il nostro obiettivo è formare arbitri che non siano semplici *utenti* di IA, ma *architetti consapevoli* di sistemi ibridi umano-artificiali capaci di produrre decisioni che siano al contempo più efficienti e più robuste.

PARTE I

FONDAMENTA TECNICHE

1. Anatomia di un Large Language Model

1.1 L'Architettura Transformer. Una rivoluzione concettuale

Per comprendere cosa sia realmente un LLM, dobbiamo partire dalla sua architettura fondamentale. I modelli pre-2017 (RNN, LSTM) processavano il testo sequenzialmente, parola dopo parola, con conseguenti limitazioni nella gestione di dipendenze a lungo raggio e impossibilità di parallelizzazione efficiente.

L'architettura Transformer risolve questi problemi attraverso il meccanismo di *self-attention*: ogni token (unità base di testo, tipicamente una parola o sub-parola) può prestare attenzione simultaneamente a tutti gli altri token della sequenza, ponderando dinamicamente la loro rilevanza.

Il Meccanismo di attenzione, matematica e intuizione

Il cuore del Transformer è la funzione di attenzione, definibile come:

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T / \sqrt{d_k}) V$$

Dove Q (queries), K (keys) e V (values) sono proiezioni lineari dell'input, e d_k è la dimensionalità delle chiavi. Questa formula apparentemente astratta nasconde un'intuizione profonda:

1. **Query e Keys.** Ogni token formula una domanda (query) e offre una risposta (key). Il prodotto scalare QK^T misura quanto ogni key risponde a ogni query
2. **Softmax.** Converte i punteggi in una distribuzione di probabilità, creando 'attention weights' che sommano a 1
3. **Values.** Le informazioni effettivamente estratte da ogni token, ponderate dagli attention weights

Esempio forense. Nella frase *La parte convenuta ha violato l'articolo 3.2 del contratto*, quando il modello processa violato, il meccanismo di attenzione assegna pesi elevati a convenuta (soggetto), articolo 3.2 (oggetto) e contratto (contesto normativo), costruendo una rappresentazione semantica complessa che cattura le relazioni giuridiche.

1.2 Dalla tokenizzazione agli embedding spaces

Prima che un LLM possa processare il testo, deve convertirlo in rappresentazioni numeriche attraverso due fasi cruciali:

Tokenizzazione

Gli algoritmi moderni (Byte-Pair Encoding per GPT, SentencePiece per Claude) scompongono il testo in unità sub-lessicali. La parola inadempimento potrebbe diventare ['in', 'ademp', 'imento'], permettendo al modello di gestire neologismi e terminologia tecnica non vista durante l'addestramento.

Embedding

Ogni token viene mappato in uno spazio vettoriale ad alta dimensionalità (tipicamente 768-12,288 dimensioni per modelli moderni). In questo spazio, la vicinanza geometrica riflette similarità semantica: 'arbitrato' e 'mediazione' avranno vettori più vicini tra loro che con 'mela'.

Implicazione pratica critica: gli embedding sono appresi statisticamente da miliardi di testi. Termini giuridici tecnici possono avere rappresentazioni imprecise se poco rappresentati nel corpus di addestramento.

Questo spiega perché Claude 3.5 Sonnet, addestrato con maggiore enfasi su testi accademici e professionali, tende a performare meglio in contesti legali rispetto a modelli addestrati principalmente su contenuti web generalisti.

1.3 Strati di trasformazione e rappresentazioni gerarchiche

I moderni LLM sono composti da decine di strati (layers) di transformer impilati. GPT-4 ha presumibilmente 120+ layers, Claude 3.5 Sonnet una configurazione simile. Ogni strato applica il meccanismo di attenzione seguito da una rete feed-forward, raffinando progressivamente le rappresentazioni:

- gli strati iniziali catturano pattern sintattici (parti del discorso, struttura delle frasi)
- quelli intermedi codificano relazioni semantiche (sinonimia, antonimia, relazioni gerarchiche)
- i finali appresentano concetti astratti e ragionamento di alto livello

Questa architettura gerarchica permette ai LLM di costruire rappresentazioni progressivamente più astratte. Quando processiamo un lodo arbitrale, gli strati iniziali identificano termini tecnici, quelli intermedi mappano le relazioni tra parti, obbligazioni e violazioni, e quelli finali costruiscono una rappresentazione del ragionamento giuridico complessivo.

2. Il processo generativo: predizione stocastica e *hallucination*

2.1 Autoregressive generation. Come gli LLM producono testo

Quando chiediamo a Claude di redigere una memoria o a GPT-4 di analizzare un contratto, il processo generativo è intrinsecamente *autoregressivo*: il modello predice un token alla volta, condizionato su tutti i token precedenti.

Formalmente, la probabilità di una sequenza di token è:

$$P(w_1, w_2, \dots, w_n) = \prod_i P(w_i \mid w_1, w_2, \dots, w_{i-1})$$

Ad ogni step, il modello calcola una distribuzione di probabilità su tutto il vocabolario (50k-200k token) e *campiona* da questa distribuzione. Il processo è stocastico, non deterministico.

2.2 Temperature e sampling strategies

La *temperature* è un iperparametro cruciale che controlla la creatività del modello:

$$P'(w_i) = \text{softmax}(\text{logits} / T)$$

- Temperature $\rightarrow 0$ Il modello diventa quasi deterministico, scegliendo sempre il token più probabile. Ideale per compiti che richiedono precisione e coerenza (es. estrazione di clausole contrattuali)
- Temperature = 1 Distribuzione originale del modello, bilanciando probabilità e diversità
- Temperature > 1 Aumenta la 'creatività' ma anche il rischio di output incoerenti o allucinatori

Best practice per arbitro: per analisi documentale e ricerca giuridica, utilizzare temperature basse (0.1-0.3). Per brainstorming argomentativo o generazione di ipotesi alternative, temperature moderate (0.7-0.9) sono appropriate.

2.3 L'*hallucination* problem in una prospettiva tecnica

Le allucinazioni degli LLM non sono bug da correggere, ma conseguenze inevitabili della loro architettura. Comprendere *perché* si verificano è essenziale per mitigarle.

Cause architettoniche delle *hallucination*

Compressione lossy. I parametri del modello (175B per GPT-3, ~200B stimati per Claude 3.5) devono comprimere l'informazione di trilioni di token di addestramento. Informazioni specifiche (date, numeri, citazioni esatte) vengono inevitabilmente distorte

Maximizzazione di verosimiglianza. Il modello è addestrato a predire token plausibili, non necessariamente veri. Una citazione giurisprudenziale inventata ma stilisticamente corretta avrà alta probabilità

Pressure autoregressiva. Una volta generato un token errato, i token successivi sono condizionati su quell'errore, propagando e amplificando l'*hallucination*

Assenza di grounding. Gli LLM vanilla non hanno accesso a fonti esterne durante la generazione. Se il modello non ha visto un caso specifico durante l'addestramento, può inventarne uno simile

Esempio critico. Se chiediamo a GPT-4 *Citami tre lodi ICC del 2023 sul calcolo del danno emergente*, il modello può generare riferimenti plausibili ma completamente inventati (ICC Award No. 28492/2023, ecc.) perché questi pattern sintattici hanno alta probabilità nel suo spazio latente, anche se i casi specifici non esistono.

3. Retrieval-Augmented Generation

3.1 Il Paradigma RAG: separare memoria da ragionamento

Il Retrieval-Augmented Generation (RAG) risolve il problema dell'hallucination separando due funzioni:

- I. **Retrieval** ➡ Recuperare fatti specifici da database esterni
- II. **Generation** ➡ Sintetizzare e ragionare sui fatti recuperati

L'architettura tipica di un sistema RAG comprende quattro componenti.

- Vector Database. Corpus di documenti (lodi, sentenze, contratti) convertiti in embeddings e indicizzati per ricerca efficiente
- Retriever. Converte la query dell'utente in embedding e recupera i k documenti più simili (tipicamente k=3-10) usando similarity metrics come cosine similarity
- Context Augmentation. I documenti recuperati vengono inseriti nel prompt del LLM come contesto
- LLM Generator. Genera la risposta basandosi esclusivamente sul contesto fornito

3.2 Embeddings semantici e ricerca vettoriale

La qualità di un sistema RAG dipende criticamente dagli embeddings utilizzati. I moderni embedding models (es. OpenAI text-embedding-3, Cohere embed-v3) mappano testi in spazi vettoriali dove la distanza geometrica riflette similarità semantica.

Esempio concreto: ricerca semantica / keyword search

Query: Responsabilità per ritardi nella consegna di opere

Keyword search tradizionale: trova solo documenti contenenti le parole esatte responsabilità, ritardi, consegn, opere

Semantic search (RAG): recupera anche documenti che discutono penalties for late completion, liquidated damages for delay, inadempimento temporale perché questi concetti hanno embeddings simili nello spazio vettoriale

Tool implementazione. Jus Mundi utilizza RAG su un corpus di 1M+ documenti arbitrari. vLex AI implementa semantic search su database giurisprudenziale multi-giurisdizionale. Entrambi permettono query in linguaggio naturale con risultati semanticamente pertinenti.

3.3 RAG / Fine-tuning / In-context learning

Esistono tre approcci principali per adattare LLM a domini specifici come l'arbitrato

Fine-tuning

- **Pro.** Integra profondamente conoscenza dominio-specifica nei parametri del modello. Miglior performance per compiti ripetitivi
- **Contro.** Costoso (richiede GPU clusters), rischio di catastrophic forgetting (perdita conoscenze generali), non aggiornabile facilmente
- **Use case.** Harvey AI probabilmente utilizza fine-tuning su corpus legale per ottimizzare performance

In-context learning

- **Pro.** Zero costi computazionali, flessibilità totale, aggiornamento istantaneo
- **Contro.** Limitato dalla lunghezza del context window, costoso in termini di token utilizzati
- **Use case.** Caricare un contratto su Claude e fare domande specifiche

RAG

- **Pro.** Scalabile a corpus enormi, aggiornabile istantaneamente, tracciabilità delle fonti, riduce hallucination
- **Contro.** Dipende dalla qualità del retrieval, può essere lento, richiede infrastruttura vector database
- **Use case.** Jus Mundi, vLex AI, CaseText - ricerca in vasti database giurisprudenziali

Raccomandazione strategica. Per la maggior parte degli arbitri, RAG offre il miglior trade-off. L'investimento in fine-tuning è giustificato solo per studi con volumi enormi e compiti altamente ripetitivi.

PARTE II

APPLICAZIONI FORENSI AVANZATE

4. Chain-of-Thought Reasoning e verificabilità

4.1 Il problema del ragionamento opaco

Uno dei limiti più insidiosi degli LLM per applicazioni forensi è l'opacità del processo di ragionamento. Quando Claude genera un'analisi contrattuale, il risultato appare magicamente, senza esplicitare i passaggi logici intermedi. Questo è inaccettabile per l'arbitrato, dove ogni conclusione deve essere giustificabile.

La tecnica del *Chain-of-Thought (CoT) prompting* (Wei et al., 2022) risolve parzialmente questo problema inducendo il modello a pensare ad alta voce, esplicitando i passaggi di ragionamento.

4.2 Anatomia di un Prompt Chain-of-Thought

Prompt standard (output opaco)

'Analizza se la clausola 3.2 del contratto allegato è stata violata dalla parte convenuta.'

Prompt Chain-of-Thought (output verificabile)

'Analizza se la clausola 3.2 è stata violata seguendo questi step: 1) Identifica l'obbligazione specifica prevista dalla clausola 3.2. 2) Estrai dai documenti le azioni concrete della convenuta. 3) Confronta sistematicamente azioni vs obbligazioni. 4) Valuta se sussiste inadempimento. Mostra il ragionamento per ogni step.'

Il prompt CoT produce output strutturati dove ogni conclusione è derivabile dai passaggi precedenti, permettendo all'arbitro di verificare la validità del ragionamento e identificare errori specifici.

4.3 Zero-shot / Few-shot CoT

Zero-shot CoT. Semplicemente aggiungere 'Let's think step by step' al prompt può migliorare significativamente il ragionamento del modello senza fornire esempi

Few-shot CoT. Fornire 2-3 esempi completi di ragionamento step-by-step guida il modello a replicare quella struttura. Particolarmente efficace per compiti giuridici standardizzati.

Esempio few-shot per valutazione testimonianze:

Esempio 1:

Testimone: A dichiara di aver visto B firmare il contratto il 15/3/2023

Valutazione: (1) Verifico se A era presente quella data → Sì, verbale riunione. (2) Verifico coerenza con altri documenti → E-mail del 16/3 conferma firma. (3) Credibilità testimone → Nessun interesse diretto. Conclusione: testimonianza affidabile.

[Inserire 2-3 esempi simili, poi la testimonianza da valutare]

Il modello replicherà il pattern di ragionamento, producendo valutazioni strutturate e verificabili.

5. Prompt engineering come competenza forense

5.1 Il prompt come atto procedurale

La formulazione del prompt non è dettaglio tecnico accessorio, ma atto sostanziale che determina la qualità dell'output. Un arbitro che padroneggia il prompt engineering acquisisce un vantaggio competitivo misurabile.

Considerate il prompt come la formulazione di un quesito peritale: la precisione della domanda determina l'utilità della risposta.

5.2 Anatomia di un prompt efficace per arbitrato

Un prompt ben costruito per applicazioni forensi dovrebbe includere:

Role definition: agisci come arbitro ICC esperto in contratti internazionali di compravendita

Task specification: descrizione precisa del compito richiesto

Constraints: basati esclusivamente sui documenti allegati. Non inventare fatti. Se informazioni insufficienti, dichiara esplicitamente la lacuna

Output format: struttura la risposta in: (1) Fatti accertati, (2) Analisi giuridica, (3) Conclusioni con citazioni precise ai documenti

Reasoning instruction: mostra il ragionamento step-by-step

5.3 Tecniche Avanzate: constitutional ai e self-critique

Una tecnica sofisticata sviluppata da Anthropic (Bai et al., 2022) è il *Constitutional AI*: incorporare principi costituzionali nel prompt che il modello deve rispettare.

Esempio per arbitrato:

'Nel redigere questa analisi, rispetta rigorosamente questi principi: (1) Imparzialità: valuta equamente le posizioni di entrambe le parti. (2) Tracciabilità: ogni affermazione fattuale deve citare fonte specifica. (3) Epistemologia: distingui chiaramente tra fatti accertati, ipotesi e congetture. (4) Proporzionalità: considera il principio di proporzionalità nella valutazione dei rimedi.'

La tecnica *self-critique* induce il modello a rivedere criticamente il proprio output:

'Dopo aver prodotto la tua analisi, riesaminala criticamente: (1) Hai fatto assunzioni ingiustificate? (2) Le citazioni sono precise? (3) Il ragionamento è rigoroso o contiene salti logici? Rivedi l'analisi correggendo eventuali debolezze.'

Questo approccio multi-pass migliora significativamente la qualità dell'output, riducendo hallucination e rafforzando il rigore logico.

6. Multi-Agent systems, il futuro della deliberazione assistita

6.1 Dal modello singolo al sistema Multi-Agente

Un'evoluzione promettente è l'uso di sistemi multi-agente dove più istanze di LLM (o LLM diversi) cooperano o competono, simulando la deliberazione collegiale.

Architettura tipica per collegio arbitrale assistito:

- **Agent A (attore):** analizza il caso dalla prospettiva dell'attore, identifica punti di forza delle pretese
- **Agent B (convenuto):** analizza dalla prospettiva della difesa, identifica debolezze delle pretese
- **Agent C (Presidente neutrale):** sintetizza le analisi precedenti, identifica consensi e dissensi, propone soluzione bilanciata
- **Agent D (Critico):** esamina la proposta di C cercando errori logici, bias, lacune probatorie

Questa architettura replica strutturalmente il processo di deliberazione collegiale, con il vantaggio di esplicitare e documentare ogni posizione.

6.2 Debate e adversarial collaboration

Una variante è il *debate framework* (Irving et al., 2018): due agenti assumono posizioni opposte e dibattono, mentre un terzo giudica quale argomento è più convincente basandosi esclusivamente sulle evidenze.

Implementazione pratica:

- Round 1. Agent Pro presenta il caso più forte per la sussistenza di inadempimento
- Round 2. Agent Contra presenta il caso più forte contro
- Round 3. Pro replica alle obiezioni di Contra
- Round 4. Contra controribatte
- Judgment. Judge agent valuta quale posizione è meglio supportata dall'evidenza

Questo processo di dialettica artificiale può rivelare argomenti e controargomenti che sfuggirebbero a un'analisi unidirezionale, migliorando la robustezza della decisione finale.

6.3 Tool: Implementazioni Esistenti

- **AutoGPT, BabyAGI.** Framework open-source per costruire sistemi multi-agente. Richiedono competenze di programmazione
- **Claude Projects (Anthropic).** Permette conversazioni multi-turn con memoria persistente, simulando parzialmente deliberazione estesa
- **GPT-4 Advanced Data Analysis.** Può eseguire codice Python, permettendo implementazione di semplici architetture multi-agente

PARTE III

GOVERNANCE EPISTEMICA E CONTROLLO

7. L'Arbitro come Human-in-the-Loop

7.1 Il paradigma HITL che va oltre la supervisione passiva

Il concetto di *Human-in-the-Loop* (HITL) è spesso frainteso come semplice revisione finale dell'output dell'IA. In realtà, HITL efficace richiede integrazione profonda e bidirezionale:

Input stage. L'arbitro formula query strategiche, definisce parametri, seleziona fonti

Processing stage. L'arbitro può intervenire mid-stream, ridirigendo l'analisi se devia

Output stage. Validazione critica, fact-checking, integrazione con giudizio professionale

Feedback loop. Gli errori identificati informano query successive, migliorando iterativamente la qualità

7.2 Framework di Validazione Epistemica

Propongo un protocollo sistematico di validazione dell'output IA per contesti arbitrali:

Livello 1: validazione formale

- Verificare coerenza logica interna: le conclusioni seguono dalle premesse?
- Identificare contraddizioni o salti logici
- Controllare completezza: tutti gli aspetti rilevanti sono stati considerati?

Livello 2: validazione fattuale

- Verificare ogni citazione: il documento menzionato esiste? Il contenuto è riportato correttamente?
- Cross-reference: confrontare affermazioni con fonti primarie
- Identificare hallucination: ci sono 'fatti' inventati plausibili?

Livello 3: validazione giuridica

- Verificare applicabilità dei precedenti citati
- Controllare interpretazioni normative contro dottrina consolidata
- Valutare se il ragionamento giuridico è persuasivo per giuristi umani

Livello 4: validazione etica

- Identificare potenziali bias nelle valutazioni
- Verificare imparzialità nel trattamento delle parti
- Assicurare che la decisione rispetti principi di giustizia procedurale

8. Adversarial testing. Red-Teaming dei sistemi IA

8.1 Il concetto di Red-Teaming

Il *red-teaming* è una pratica di sicurezza informatica in cui un team tenta intenzionalmente di compromettere un sistema. Applicato agli LLM, significa testare sistematicamente vulnerabilità, bias e failure modes.

Per contesti arbitrari, red-teaming significa porre domande progettate per:

- indurre hallucination:** 'citami il lodo ICC 2024/12345 dove si discute X' (il lodo non esiste)
- esporre bias:** presentare lo stesso caso con parti di nazionalità diverse, verificare se le conclusioni cambiano
- testare robustezza:** formulare query ambigue o contraddittorie, verificare se il modello richiede chiarimenti o procede incautamente
- verificare boundaries:** chiedere al modello di violare principi etici (es. suggerire strategie per occultare prove), verificare se rifiuta appropriatamente

8.2 Casi di Studio: failure modes documentati

Caso 1- Il Problema delle 'Citazioni Zombie'

Nel 2023, un avvocato americano (caso Mata v. Avianca) presentò memorie contenenti citazioni giurisprudenziali generate da ChatGPT. I casi erano completamente inventati ma stilisticamente perfetti. Il giudice sanzionò l'avvocato.

Analisi tecnica del failure: GPT-3.5 (il modello utilizzato) ha pattern statistici molto forti per citazioni legali (formato 'Plaintiff v. Defendant, Volume Reporter Page (Court Year)'). Quando richiesto di citare precedenti inesistenti, il modello genera istanze plausibili di quel pattern anche se i casi non esistono nel training data.

Mitigazione: usare sistemi RAG che possono citare solo da database verificati. Mai fidarsi di citazioni senza verifiche incrociate con fonti primarie.

Caso 2- Bias geografico in valutazioni contrattuali

Test condotto da ricercatori (Ammanabrolu et al., 2023): stessa clausola contrattuale ambigua presentata a GPT-4 con parti di diverse nazionalità. Bias sistematico verso interpretazioni favorevoli a parti USA/UK vs altre giurisdizioni.

Causa: over-representation di case law angloamericano nel training data. Il modello ha imparato bias impliciti presenti nella giurisprudenza.

Mitigazione. Testare sistematicamente con parti di nazionalità diverse. Usare Constitutional AI con principi di imparzialità espliciti.

9. Tracciabilità, auditabilità e governance operativa

9.1 Il problema del non-determinismo

Una sfida fondamentale nell'uso forense di LLM è il non-determinismo: lo stesso prompt può produrre output diversi in esecuzioni successive a causa del sampling stocastico.

Questo solleva questioni di auditabilità: se un'analisi IA contribuisce a un lodo arbitrale, come documentiamo esattamente cosa è stato chiesto e cosa ottenuto?

9.2 Protocollo di documentazione completa

Propongo un protocollo standard per documentare l'uso di IA in procedimenti arbitrali:

- metadata del modello.** Nome preciso (es. 'claude-3-5-sonnet-20241022'), versione, data di utilizzo
- prompt completo.** Salvare il testo esatto della query, inclusi documenti allegati e context
- parametri.** Temperature, top_p, max_tokens, seed (se disponibile per reproducibilità)
- output completo.** Risposta integrale del modello, non solo le parti utilizzate
- validazione.** Documentare il processo di verifica: quali fatti sono stati controllati, quali fonti consultate
- utilizzo finale.** Quali parti dell'output sono state incorporate nel lodo, quali modifiche apportate

Tool implementazione: piattaforme come Harvey AI includono built-in audit logging. Per uso ad-hoc di Claude/GPT, creare manualmente un AI Usage Log in formato strutturato.

9.3 Disclosure e trasparenza verso le parti

Una questione giuridica aperta è se e come divulgare l'uso di IA alle parti. Diverse istituzioni arbitrali stanno elaborando linee guida:

- **Approccio massimalista.** Disclosure completa preventiva di ogni utilizzo di IA, con possibilità per le parti di obiettare
- **Approccio minimalista.** Disclosure solo se l'IA ha contribuito sostanzialmente alla decisione
- **Approccio procedurale.** Disclosure in atti procedurali preliminari, permettendo alle parti di concordare protocolli

Raccomandazione. In assenza di consensus internazionale, propendo per disclosure proattiva generica ('Il tribunale può avvalersi di strumenti di IA per analisi documentale e ricerca giuridica, sempre soggetti a validazione umana completa') nelle Procedural Order iniziali, con disclosure specifica in memoria finale se l'IA ha contribuito materialmente.

VERSO UN'INTELLIGENZA IBRIDA

Gli Large Language Models non sono intelligenza artificiale nel senso fantascientifico di macchine pensanti. Sono strumenti statistici estremamente sofisticati che hanno imparato a modellare la distribuzione del linguaggio umano attraverso l'esposizione a porzioni significative della conoscenza umana scritta.

La loro forza non sta nel sostituire il ragionamento umano, ma nell'*amplificarlo*: processare volumi di informazioni che sarebbero impraticabili per individui singoli, identificare pattern che potrebbero sfuggire, generare ipotesi da validare, strutturare ragionamenti da verificare.

La loro debolezza è speculare: mancanza di grounding nella realtà, tendenza all'hallucination, assenza di comprensione genuina, bias sistematici, non-determinismo. Questi non sono difetti transitori da correggere, ma conseguenze architetture intrinseche.

Il futuro dell'arbitrato non è IA contro umani, bensì **intelligenza ibrida**: sistemi in cui arbitri umani e sistemi artificiali collaborano, ciascuno contribuendo le proprie capacità distintive. L'arbitro fornisce giudizio, contestualizzazione, responsabilità etica e legale. L'IA fornisce elaborazione su scala, ricerca esaustiva, identificazione di pattern, generazione di ipotesi.

Ma questa simbiosi richiede comprensione tecnica profonda. Un arbitro che usa l'IA come black box magica è pericoloso quanto uno che la rifiuta dogmaticamente. La competenza del futuro è saper *orchestrare* intelligenza umana e artificiale, conoscendo le capacità e i limiti di entrambe.

L'arbitrato è per sua natura conservatore, e giustamente: la stabilità e prevedibilità delle istituzioni giuridiche è valore fondamentale. Ma conservatorismo non significa immobilismo. Le tecnologie di LLM sono qui, sono potenti, e saranno sempre più pervasive. La domanda non è se usarle, ma *come* usarle responsabilmente, efficacemente, e in modo che rafforzi piuttosto che compromettere l'integrità del processo arbitrale.

Questo è il compito che ci attende.

BIBLIOGRAFIA SELEZIONATA

- Vaswani, A., et al. (2017). "Attention Is All You Need." *Advances in Neural Information Processing Systems*, 30.
- Bai, Y., et al. (2022). "Constitutional AI: Harmlessness from AI Feedback." *ArXiv preprint arXiv:2212.08073*.
- Wei, J., et al. (2022). "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models." *Advances in Neural Information Processing Systems*, 35.
- Irving, G., et al. (2018). "AI Safety via Debate." *ArXiv preprint arXiv:1805.00899*.
- Lewis, P., et al. (2020). "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks." *Advances in Neural Information Processing Systems*, 33.
- Ammanabrolu, P., et al. (2023). "Do LLMs exhibit human-like response biases? A case study in survey design." *Transactions of the Association for Computational Linguistics*.
- Anthropic. (2024). "Claude 3.5 Sonnet: Model Card and Evaluations." *Technical Report*.
- OpenAI. (2024). "GPT-4 Technical Report." *ArXiv preprint arXiv:2303.08774*.
- ICC Commission on Arbitration and ADR. (2023). "Guidance Note on the Use of Artificial Intelligence in International Arbitration." *ICC Publication*.
- LCIA. (2024). "Discussion Paper: AI and Arbitration." *LCIA Notes*.